

Lesson 02

控制台界面

CLI - Command Line Interface

程序参数

Program Arguments

02_cli

案例

```
ls -a
ls Documents
ls -a Documents
```

程序参数形态

原生程序：

```
<program> <args>
ls Documents
```

语言启动的程序：

```
<language> <program> <args>
java zzax.Driver zhaozhe
node dice.js --style 4
```

Java 获取 程序参数

```
1 public class Driver {
2     public static void main(String[] args) {
3
4     }
5 }
```

参数规范

Option：

```
-a      option short version
--add   option long version
```

实战

建造一个 支持程序参数的 计算器

IntelliJ 内部配置 程序参数

环境变量

Environment Variables

什么是

系统脚本的变量

这个变量会传到所有的启动的程序里

查看现有变量的指令

Mac:

```
env
```

Windows:

```
set
```

添加新变量

Mac:

```
export HEY=123
```

Windows:

```
setx HEY 123
```

Java 代码查看现有环境变量

03_env

```
System.getenv();
```

用途

给程序进行传参

更像程序配置

IntelliJ 中叠加环境变量

程序参数 vs 环境变量

程序参数	用于传输数据
环境变量	用于传输宏观配置 以及需要保密的信息

包管理

Package Management

背景

基于 别人的代码

我们不可能所有的代码都自己写

```
str.indexOf("ZZAX");  
System.out.println(str);
```

大型 SDK

Java SDK 里面已经有很几乎所有你需要用的到类

但是依然有一些设计过时，或者有些需求 SDK 中并没有给出比较好的处理工具

三方贡献 代码

比如 JS / Python 官方并没有很健全的 SDK。
很多开发者 自发自愿 为 JS 去 写 通用代码

如何分享代码

1. 直接复制粘贴共享代码
2. 使用人工导包共享代码
3. 使用包管理工具共享代码

直接复制粘贴 共享代码

问题

1. 引用时过于复杂
2. 代码更新时，需要重新复制粘贴代码
3. 源代码暴露

使用人工导包 共享代码

Jar 包

一个压缩包，里面有编译后的 .class 文件
但是不运行

生产部分代码

新建项目：

03_jar

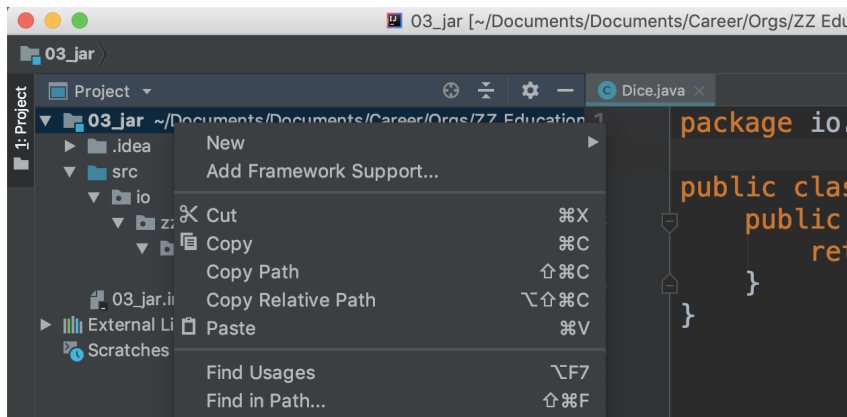
```
1 public class Dice {  
2     public static int roll(int max) {  
3         return (int)(Math.random() * max) + 1;  
4     }  
5 }
```

导出

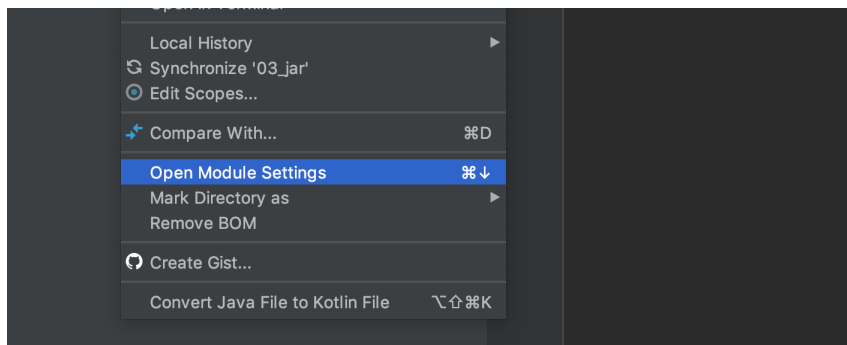
配置 jar 导出：

左上角项目点击 右键

看到



往下 看到



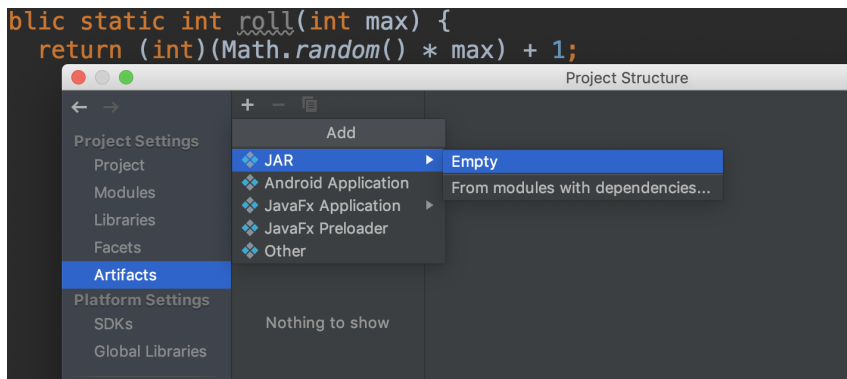
点击 **Open Module Settings**

你会看到 一个弹出的窗口

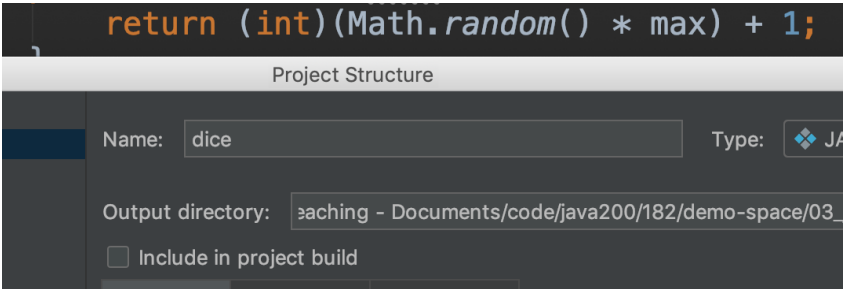
左侧点击 **Artifacts**

找到 **+**

点击 **+** > **Jar** > **Empty**

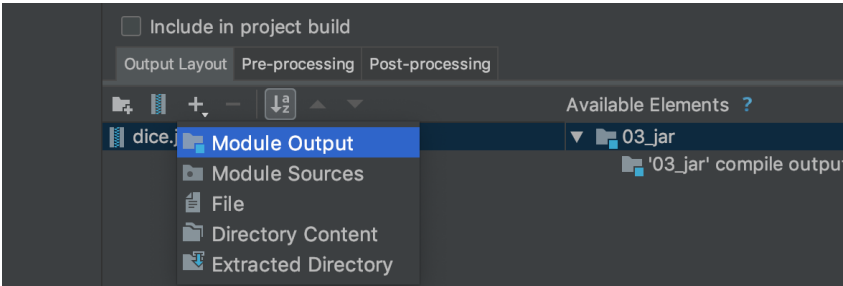


看到



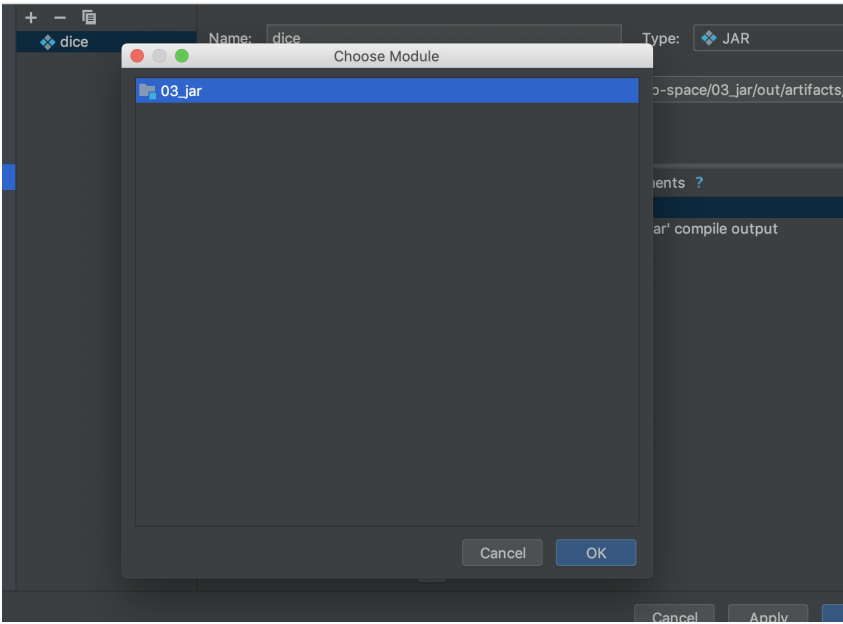
Name 处 填写 dice

找到



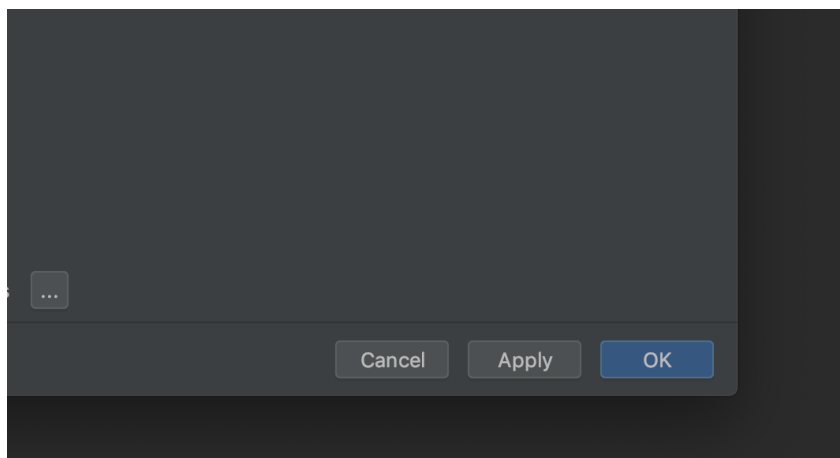
点击 > module output

看到



点击

看到



点击 **ok**

导出：

点击 菜单下的 **Build** > **Artifacts** > **Build**

检查 out/artifacts/dice 下 是否有 jar 包

导入

新建项目：

04_lib

1. 拷贝 jar 文件：

项目名字部分 点击右键

点击 **New** > **Directory**

看到窗口

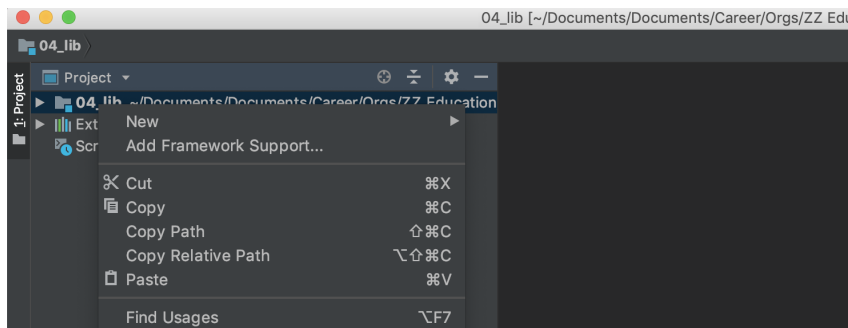
填写 lib

点击 **Ok**

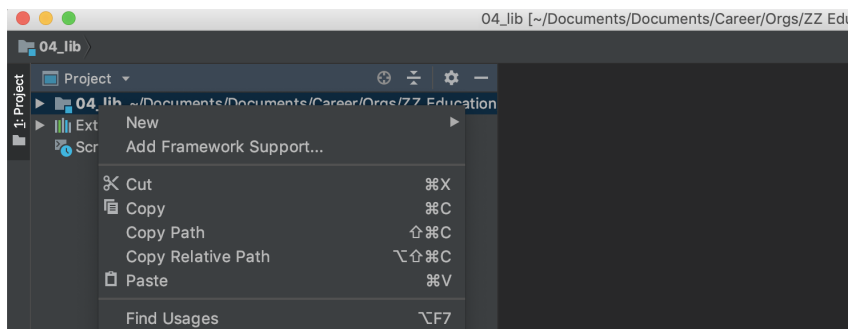
将 jar 文件 拷贝到 lib 文件夹下

2. 导入 jar：

项目 点击 右键

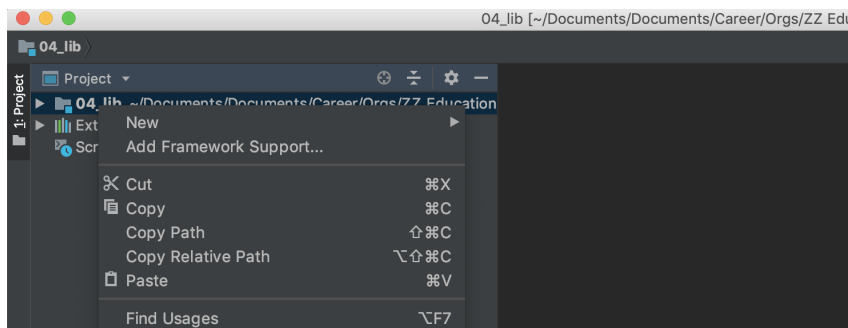


往下看



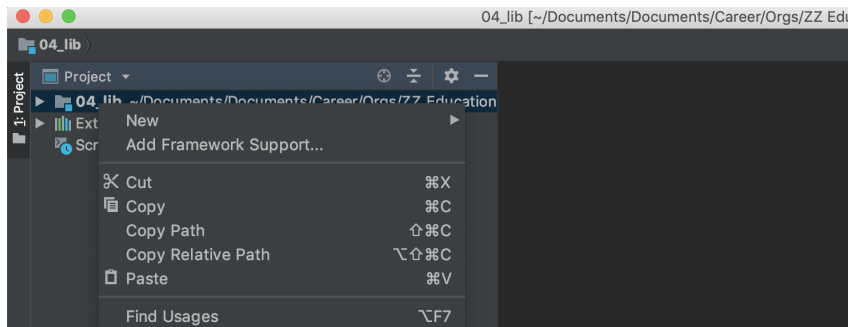
点击 **Open Module Settings**

看到



确认目前为 **Modules** > **04_lib** > **Dependencies**

点击 下方 **+** > **jar**



找到 在 lib 文件下的 dice.jar 文件

点击 **Ok**

点击 **Ok**

使用导入代码

```
1 import io.zzax.game.Dice;
2
3 public class Lottery {
4     public static void main(String[] args) {
5         System.out.println(Dice.roll(6));
6     }
7 }
```